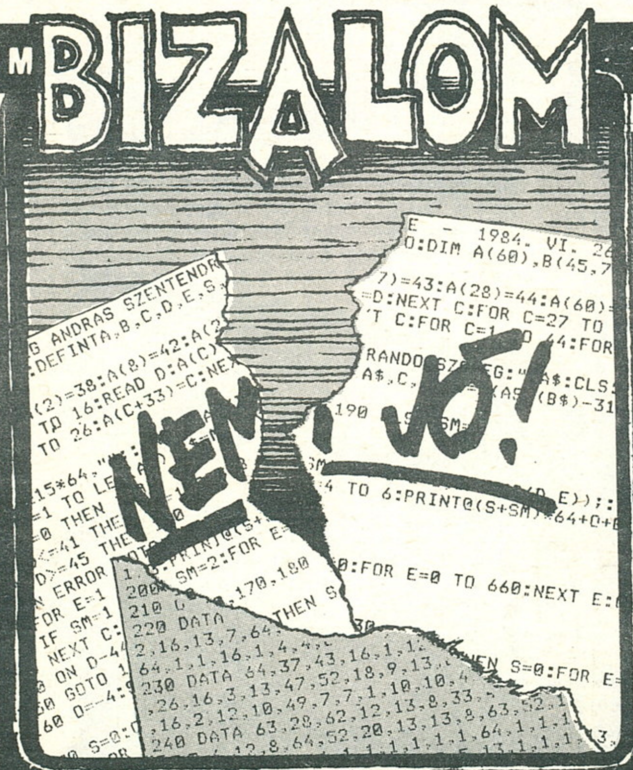
BIT-LET

ötlet SZÁM · 1985 MÁJUS 20



Barátom gm-ezik. Nemrégen pedig az általa is jegyzett játékszervező gm olyan országos vetélkedőt szervezett, amelynek lebonyolításához, a játékban részt vevő nagy számú csapat pontjainak összesítéséhez nélkülözhetetlen volt a számítógép. S mert egy jól szervező szervező semmit sem bíz a véletlenre, a játékszervező gm e számítógépes program megírására komoly pénzért szerződtetett egy profit. Aki ugyan lényegesen többet kért el e munkáért, mint az ismerősöm által ismert sok tehetséges, lelkes fiatal programozó, de a pénzért profi teljesítményt ígért és nyújtott is.

A program elkészült, a programot tesztelték, majd kijavították, majd újra tesztelték, újra kijavították, s mert tudjuk, hogy hibátlan program nincs, mindez addig ismétlődött, mígnem eljött az idő, amikor már nem lehetett tovább tesztelni és javítani, hanem el kellett fogadni, hogy a program ugyan tökéletes soha nem lesz, de most már véglegesnek kell tekinteni. A programot a 19 megyeközpontban és a fővárosban egyidőben zajló vetélkedőben használták, így hát el kellett belőle készíteni megyénként 3-3 példányt és át kellett adni a programot kezelő, használó számítógépeseknek.

S most következik mindaz, amiért a szerkesztő ezt a saját „magánszférájában" szerzett tapasztalatot úgy érezte, érdemes közreadni. Az ország húsz pontjáról érkezett számítógépesek a kész programot képtelenek voltak fölhasználóként elfogadni, s ahelyett, hogy arra figyeltek volna, hogy hogyan lehet legoptimálisabban, leghasznosabban munkájukban fölhasználni azt, inkább arra próbálták rávenni a program jelen lévő szerzőjét és a programot megrendelő játékszervezőket, hogy tegyék lehetővé, hogy a programot ki-ki ízlése szerint változtassa, alakítsa. A program lefordított állapotban került a lemezekre, s a leendő fölhasználók egy részét ez kifejezetten sértette, közülük többen követelték, hogy kapják meg azt lefordítatlanul is, hogy tetszésük szerint alakíthassanak benne.

Tipikusnak éreztem a történteket. Tipikusnak abban az országban, ahol a programozói szemlélet olyan, hogy „jobb, ha mindent magad írsz meg, jobb, ha csak magadban bízol, amelyik programot más írt, az csak vacak lehet". A szemlélet létezik, a szemlélet programozónemzedékek nevelésének is egyik fő motívuma, másként hogy is lehetne, hogy már fiatal, kezdő programozók sem bíznak mások kész szellemi termékeiben, ha azt a terméket programnak hívják.

Kár... Kár, mert ez nagy pénzek kidobásához vezet szerte az országban. Vállalatok, irodák tucatjai vesznek gépet mindenféle program nélkül, s inkább maguk barkácsolnak, lopnak, mintsemhogy a készen kapható programokat használnák az adott helyen felmerülő speciális igények szerint.

Az, hogy egy kezdő programozó minden programról tudni szeretné, hogy hogyan működik, még csak érthető. Olyan ez, mint a kisgyerek törekvése, hogy babának, embernek, állatnak egyaránt kiszedje a szemét, lecsavarja a fülét. A fura, hogy míg a gyerek néhány év alatt – megismerkedve az őt körülvevő világgal – leszokik erről, tudomásul veszi, hogy a dolgok nem szétszedésre valók, addig a programozók jelentős része még évek múltán sem képes belenyugodni, hogy lehetnek programok, amelyek működését nem ismerheti meg, s amelyeknek csak felhasználója lehet.

Valamilyen módon gyógyírt kellene találni erre a betegségre. Sőt, gyógyírt kellene találni arra is, hogy az ország különböző részein ne kelljen tízszer, százszor megírni ugyanazt a programot. Ha a mikroszámítógépeket használó vállalatok szakemberei képesek lennének elfogadni, hogy nem kell minden programot belülről ismerniük, akkor onnan már csak egy lépés lenne, hogy vásárolni is akarjanak mások által írott programokat. S ebben, a hazai készítésű programok információáramlásában szívesen segítene lapunk is. Börzerovatunk ma még nincs, de éppen lehetne. Egy ilyen rovatban ki-ki közölhetné eladó programjainak nevét, a program kétsoros felhasználói leírását, a becsült eladási árat és azt a címet, ahol bővebb információt lehet kapni a programról.

Hogy ilyen rovat lesz-e a BIT-LET-ben, ez elsősorban Önökön, olvasókon, programozókon múlik. Tartok tőle azonban, hogy először nem a levélírás iránt érzett ellenszenvüket kell legyőzni, hanem az egymás iránti bizalmatlanságot. Ha ez sikerült, küldjék el első ajánlataikat, kíváncsian várja őket a szerkesztő:

Angyalosi László

BELÜLRŐL

26 ***Hiroldal*** – megtekinthető benne egy billentyűzet, amely gnómnak látszik, de legkevésbé sem az!

28 ***Vallató hozzászólás – VC 20*** – úgy tűnik, VC-es körökben nem aratott tetszést vallatónk

30 ***Vallató hozzászólás Primó*** – egy olvasó védelmébe veszi a Primi-t

30 ***Szoftvertötletek*** – egy új rovat első anyaga a HT véletlenszám-generátoráról

31 ***Programajánlat*** – egy scrollozó program Spectrumhoz

32 ***A HT grafikáról kezdőknek és haladóknak*** – segítség és összefoglalás mindazoknak, akik így vagy úgy, de rajzolgatnak a HT-n

34 ***A RAM és a képernyő*** – egy cikk, amely elsősorban a haladóknak és elsősorban a Spectrumosoknak szól

36 ***Sorvezető*** – ismétlés a tudás anyja címmel a HT billentyűzetével kapcsolatos tudnivalókat foglalja össze Zátonyi Sándor

39 ***Posta*** – amelyben egy olvasó fölkínál egy magyar nyelvű hibaüzenetet tudó kis szoftvert a HT-re...

40 ***Micolor nyerő*** – íme a második feladat, s némi visszatérés a Primó-nyerő megfejtéseire

HÍROLDAL

Datapoint

Datapoint számítógépet vettek használatba Zalaegerszegen az IBUSZ-irodában. A számítógéppel lehetőség nyílik a megyeszékhely és a főváros IBUSZ-irodái közötti összeköttetésre. Elsősorban a társasutazások és a szobafoglalások esetében alkalmazzák sikeresen az új gépet.

Hotel Computer

Új szolgáltatás bevezetésével kísérletezik az amerikai Hyatt szállodalánc. Chicagói szállodájában IBM személyi számítógépeket kölcsönöz szállóvendégeinek, és így azoknak üzleti útjaikra nem kell magukkal hozniuk saját gépüket. A PC-ket óránként hat dollárért vehetik igénybe az ott lakók. Ha a kísérlet sikeres lesz, akkor a Hyatt egész hálózatára kiterjeszti az új szolgáltatást.

Képernyőhír

A Zenith Data Systems olyan notesz méretű hordozható számítógépet tervez piacra dobni, amelynek folyékony kristályos képernyőjét hátulról világítják meg. A 16 font súlyú 175-ös típusú gép egy vagy két 5,25 inches floppy meghajtóval 256 kbyte RAM-mal, egy 80 C 88-as processzorral rendelkezik. Ára 3000 dollár lesz. A gép igazi újdonsága a hátsó megvilágítású folyékony kristályos képernyő. Erre azért van szükség, mert a tapasztalatok szerint szobai megvilágítás esetén, bizonyos szögből a normál folyékony kristályos képernyő olvasása szemrontó és kényelmetlen.

Master

A Contral Data olyan Winchester típusú háttértárakat szállít az IBM PC AT-hez, amelyek beilleszthetők a PC AT dobozába. A Storage Master 630-as típus méreteiben igazodik az IBM személyi számítógépéhez. Kettő helyezhető el belőlük, méretük egyenként 30 Mbyte. Tipikus elérési idejük 30 millisecundum, amely kb. 25%-kal jobb az eredeti IBM meghajtó egységek megfelelő adatánál. A Winchester adatmodul az eredeti IBM vezérlőt használja. Kiegészítő szoftver szükséges ahhoz, hogy a PC–DOS 3. 0 lehetőségeit a felhasználó kihasználhassa. Ez a szoftver lehetőséget ad a lemez logikai felosztására és formatálására is. A Winchester lemezekhez gyors mágnesszalag kazetta is rendelkezésre áll az adatok mentéséhez. A Winchester lemez ára 2145 dollár.

Termálnyomtató

A General Electric bejelentette az új Model 3–8100 típusú, terminálos, levél minőségű sornyomtatóját 300 dollárért.
A Model 3–8100-as típus Centronics csatlakozóval rendelkezik, két sebességgel tud működni, 50 karakter/másodperc a normál, 25 karakter/másodperc a levél minőségű írássebessége.

250 után 260

A Hewlett Packard a 250-es sorozat után piacra dobja a 260-as sorozatot. Ezek több felhasználós személyi számítógépek lesznek. A legkisebb modell a HP 260/15-ös 256 kbyte memóriával, 15 megabyte-os merev lemezes tárolóval és mikrofloppy egységekkel van ellátva. Ára 10 500 dollár.
A legerőteljesebb modell, a HP 260/55 már 18 felhasználót tud kiszolgálni 55 megabyteos merev lemezes tárolóval és az állományokat mentő szalagkazettákkal.

Las Vegasi hírek!

Az Atari hat, a Commodore két új kommersz személyi számítógéptípust mutatott be a Las Vegas-i Consumer Electronics kiállításon januárban.
Az Atari hat újdonsága közül kettő az 530 ST és az 520 ST ragadta meg különösebben a vásárlók figyelmét. Mindkét gép Motorola 68 000-es központi egységet alkalmaz és alapszoftverként a CP/M–68 K operációs rendszert valamint a „Gem" nevű (Graphics environment manager, grafikus környezetszervező), Digital Research-től származó grafikus szoftvert használ. Mindkét gépnek 198 K ROM-ja van.
A 130 ST alapgépe 128 K, az 520 ST 512 K RAM-mal rendelkezik, áruk ennek megfelelően 399, illetve 699 dollár. A csatolható perifériák között találhatjuk a 3,5 inch-es mikrofloppy-t csakúgy, mint a különböző merevlemezes háttértárakat.

A Commodore új gépek a következők:
C 128
Az alapgépben két processzor van, egy 6502-es és egy 280-as. Operatív memóriája 128 K, ára 300 dollár. A gép mind a Commodore 64-gyel, mind a CP/M operációs rendszerű gépekkel kompatibilis.
Folyékony-kristály-képernyős, hordozható
Ez a gép szintén 6502 alapú. A RAM mérete csak 32 K, de a ROM elég nagy méretű: 96 K. Itt foglal helyet a „lapméretű" gépek szokásos szoftver választéka plusz egy spreadsheet program. A képernyő mérete 16x80, folyékonykristályos. A gép ára még nem ismert.
Az új modellek megjelenése tovább élesíti a piaci versenyt. Az Atari 68 000-es gépei például eladhatatlanná tehetik a Sinclair QL-jét az Egyesült Államokban.

Spéci tv!

A General Electric egy színes és egy feketefehér tv-készüléket kezd árusítani, amelyeket kifejezetten számítógép-monitorként való alkalmazásra terveztek. A tv-készülékek 80 karaktert jelenítenek meg soronként, alkalmasak összetett videójelek fogadására és szabványos antennacsatlakozóval is rendelkeznek. A színes készülék ára 490 dollár, a feketefehér készüléké 130 dollár. A tv-készülékekkel egyidőben a General Electric egy akusztikus modemet is bejelentett 120 dollárért.

Inforum

Amikor a hartfordi Travelers biztosító iroda adminisztrátorainak szükségük van arra, hogy Norsrossban vagy Maitlandben levő kollégáikkal konzultáljanak, egy új eszköz, az INFORUM telekonferencia rendszer segíti az adatok, képek és ötletek cseréjét. Léteztek már olyan videóhálózatok, amelyek segítségével tárgyalásokat lehetett lebonyolítani. Hogy ezek a rendszerek nem tudtak elterjedni, azt a videókészülékek magas ára indokolja.
Az American Video Teleconferencing Corporation fejlesztőié az érdem, hogy fölismerték, ezeken a megbeszéléseken az információ nagy része korántsem videócsatornákon továbbítható leghatásosabban. Ezen ötletből fejlődött ki az INFORUM.
A rendszer három fő eleme: számítógéphálózat, hang- és videócsatorna. Ez utóbbi csak kiegészítő elem; mindössze állóképeket tud közvetíteni. Minden tárgyalóteremben két kamera és két monitor van elhelyezve. Egy kamera a résztvevők, egy pedig az esetleges segédanyagok képét közvetíti, mialatt természetesen folyamatos a hangösszeköttetés. Az egész rendszer működését IBM PC-k fogják össze, ezekhez úgynevezett kommunikációvezérlő, egy videócsatorna és egyéb

hardver berendezések csatlakoznak, továbbá a helyi iroda adatfeldolgozó IBM 308X számítógépe. A rendszer valódi értéke – és újdonsága – ebben az adathálózatban rejlik. Tetszőleges, lemezen tárolt anyag (táblázat, rajz, szöveg stb.) behívható az állomás PC-jének memóriájába, onnan pedig továbbítható az összes többi helyszínre. Itt azután elvégezhetők módosítások (pl. újra átszámolhatók és helyesbíthetők táblázatok adatai).
Az összes berendezést egy „teremfelügyelő" kezeli egy érintés-vezérelt képernyőn levő menü segítségével. Így indíthat több helyszínes hívásokat, kiválaszthatja és eloszthatja a konferencia képeit vagy továbbíthat a számítógépektől nyert adatokat, rajzokat.
Az INFORUM 1984 végén kezdte meg a működést a három helyszínes Travelers rendszerben. A berendezés tíz helyszínig bővíthető, s a Travelers máris tervezi ezeken a helyszíneken a videókészülékek felszerelését. A rendszer berendezései, azok beüzemelése együttesen 143 ezer dollárba kerülnek helyszínenként. Ehhez külön költségként járul hozzá a rendszert vezérlő programrendszer bérleti költsége. A teljes ár és a különköltségek viszont nem teszik ki a hagyományos videóláncok árának egytizedét.

Micro-suli

Nyári programozási diáktáborról kaptunk hírt a műszaki egyetemistáktól. Jún. 17–aug. 30-ig 11 héten keresztül szerveznek programozási tanfolyamokat Budapesten hetes turnusokban, napi 4 vagy 8 tanórában 20–25 Ft-ért óránként.
Commodore, Spectrum számítógépeken kezdő, haladó BASIC, assembler, valamint PASCAL FORTH és MICRO-PROLOG tanulására lehet jelentkezni!
Cím: MICRO-STÚDIÓ: 1536 Bp. Pf. 323. T.: 460-832.

Hamisítványok

Az IBM PC AT alig „melegedett meg" az első vevőknél, megjelentek a piacon az utánzók. Az egyik ilyen forgalmazó a Kaypro Corp., aki a Kaypro 286 i nevű termékét kezdi árusítani a következő összeállításban: Futel 80286-os processzor, két 1–2 megabyte-os floppy mágneslemez egység, 512 kbyte RAM, amely bővíthető 640 kbyte-ra, további bővítő kártyákkal a memória mérete 15 megabyte RAM lehet. A számítógép funkcionálisan kompatibilis az IBM PC AT-val. Az alapgéppel szállított szoftver: Word Star, Mail Merge, Inforster, Calc Star és a GW Basve. A számítógép ára 4550 dollár.

Nyelvújdonság

A Systems Management Associates nevű cég új magas szintű programozási nyelvet fejlesztett a Commodore 64-es személyi számítógépekhez. A fordítóprogram neve PROMAL (*Pro*grammer's *M*icro *A*pplication *L*anguage).
A nyelv lehetővé teszi olyan feladatok megoldását magas szintű nyelven, amelyeket eddig assembler szintű nyelven kellett megoldani. A teljes képernyős szerkesztési lehetőség, az adatállományok könnyű kezelhetősége, a karakterláncokra vonatkozó függvények csábítóvá teszik a programnyelvet. A gyártó szerint a nyelv 70–200%-kal gyorsabb, mint a Basic, FORTH vagy Pascal. A fordítóprogram ára kb. 50 dollár.

Miskolci hír

A miskolci Földes Ferenc Gimnázium fennállásának 425. évfordulója tiszteletére ünnepségsorozatot szervezett. Ennek keretében volt és jelenlegi diákjainak számítástechnikával kapcsolatos munkáiból bemutatót rendezett, valamint egy vitadélutánt az eddigi eredményekről és az oktatás további lehetőségeiről.
Minderről az iskola egy összefoglaló kiadványt tervez megjelentetni, amely kapcsán lapunkban erre szeretnénk visszatérni.

Cukubai

A Tokio melletti Cukubában szeptember közepéig nyitva tartó világkiállítás számos világszenzációval szolgál. A legtöbb elektronikus különlegességet a házigazda japánok produkálják. Egyik büszkeségük a kottaolvasó, elektronikus orgonán játszó robot. A másik robotuk miközben beszélget a látogatókkal, pillanatok alatt megfesti a jelentkezők portréját. Szenzációs eredményeket mutatnak be a fordítógépek kategóriájában is. Japánról angol, francia és német nyelvre fordító automatikus masinájukat már csak az a távközlési eszközökkel is egybeépített berendezésük múlja felül, amellyel például egy angol és egy francia személy egymás nyelvének ismerete nélkül, a fordítógép segítségével közvetlenül telefonálhat egymással.



A képet nem a fotós rontotta el és nem is gyűrődött össze, ez valóban egy billentyűzet. Az új Malcrom típusú klaviatúrán, mint gyártói állítják, negyedannyi idő alatt tanulható meg a szabályos gépelés, mint a hagyományos billentyűzeten. Ami még ettől is lényegesebb, az új forma kiküszöböli az ujjízületeket károsító természetellenes mozdulatokat és a kényelmetlen csuklótartást, ami a nyakas vállfájások fő okozója. Az eddigi vizsgálatok szerint azokon a helyeken, ahol a Malcrom billentyűzetet már alkalmazzák, a munka termelékenysége 50%-kal nőtt, ugyanakkor kevesebb a hiba és a dolgozóknak minden ízületi problémája ami a hagyományos billentyűzetet használó alkalmazottaknál már elfogadott munkahelyi ártalom. Az új billentyűzetet egyelőre az IBM PC és a BBC gépekhez gyártják.

Reklám zene

A mikroáramkörökkel megoldott zenés karácsonyi, újévi üdvözlőlapok sikeres debütálását követően újabb elektronikus csodát ismerhet meg a világ, a zenélő újsághirdetést. A közelmúltban a Le Point párizsi hetilap olvasóit váratlanul érte, hogy ujjuk érintésére az IBM amerikai mammutvállalat fényképes számítógép-hirdetése egy karácsonyi dalt játszott el. A hangos hirdetés magyarázata, hogy az újság papírjába egy szupervékony IBM nyomtatott áramkört építettek be.



Tisztelt Szerkesztőség!

Önök két öngólt rúgtak a VIC–20-as számítógéppel kapcsolatban! Az egyik az, hogy csak az „erőszaknak engedve", éves késéssel tűzték napirendre a gép tesztelését. A másik az, hogy ÍGY!

Nem akarom megbántani a Vallatóban résztvevőket, de a többségükről a véleményük (tévedéseik) alapján kitűnik, hogy néhány óránál többet nem ültek a VIC–20 előtt. Ilyen mélységű ismeretek alapján pedig nem illik megalapozott véleményt mondani egy gépről (aki nem tud arabusul...).

Felsorolnék néhány dolgot, amely kétségessé teszi a kapott osztályzatok egy részének megalapozott voltát.

1. kin. Miért változott meg az eddigi gyakorlat, hogy a külföldi árát osztályozzák? A vallatás ezzel a nem teljesen fair változtatással kezdődik.

(Ennek oka egyszerű. A legutóbbi nyugati leértékelés után olyan olló keletkezett a kinti és itthoni ár között, amely irreálissá tette volna a kinti végkiárusítási ár osztályzását. S egyáltalán nem reális egy ilyen dömpingárat osztályozni, amely nem tükrözi a gép kinti árfekvését sem! – A szerk.)

2. kin. Ha a géphez minden periféria kapható (és kapható!), akkor miért kapott a gép hét 3-ast is?

3. kin. A leírttal ellentétben 22 karakter írható egy sorba! A Super Expander nevű programmal BASIC-ből is egyszerűen kezelhető a nagy felbontású grafika!

4. kin. A Super Expanderrel a hanggenerátor is egyszerűen kezelhető. A Junoszty tv-n a hang soha nem is fog megszólalni rendesen, mivel a gép modulátora PAL rendszerű, ezért csak kétnormás (PAL-SECAM) tv-vel kapunk színes képet és hangot! Ezt még egy Vallatónak is illene tudni!

5. kin. A kazettás tárolás sebessége ugyanakkora, mint a piacon kapható gépek 99%-ának! De ha ez valakinek nem elég, kapható olyan cartridge, amely ezt a sebességet a tízszeresére növeli! A kazettás tárolást huzamosabb ideje használók sem igen tudják, mi az, hogy LOAD ERROR! Valószínű, hogy a SPECTRUM, PRIMO, HT stb. felhasználói irigykedve sóhajtanak fel ennek hallatán!

6. kin. A hümmögés inkább azért lehetett, mert a Vallatók nem tudták, hogyan lehet gépi kódban programozni a VIC–20-at. Mindenféle tárbővítés nélkül a VICMON nevezetű program segítségével nagyszerűen lehet gépi kódban programozni a gépet! Méghozzá úgy, hogy a bebillentyűzött memonikot rögtön ellenőrzi, lefordítja és elhelyezi a memóriában. Ideális eszköz egy kezdőnek, hogy megtanuljon gépi kódban programozni!

7. kin. A tápegység a „buherálók" ellen, életvédelmi szempontból van műgyantával kiöntve. Egyébként is miért kellene javítani? Nem szokott elromlani!

8. kin. A billentyűzetre kapott osztályzattal kapcsolatban csak annyit, hogy aki ebben a kategóriában ennél jobbat akar, annak azt kívánom, hogy egész életében csak ZX81 billentyűzetet nyomkodjon!

9. kin. A felhasználói kézikönyv valóban nem túl bőbeszédű, viszont létezik a VIC–20 programozói

kézikönyv, amely sokkal részletesebb, megadja a memóriatérképet, sőt jó néhány interpreter rutin leírását, kezdőcímét is megadja, példákkal illusztrálva a használatot.

10. kín. Aki dolgozott már SCREEN EDITOR-os géppel, az tudja, hogy ennél kényelmesebb mód a szerkesztésre nincs. Nem éppen hízelgő a vallatókra, hogy a 4 (négy!) editorgomb használatába belebonyolódtak!

11. kín. A gép programnyelve teljes egészében megegyezik a Commodore 64 gépével, ezért kicsit csodálkozom, hogy ennyire rossznak találtatott. Sajnos nem tudtam rájönni arra, mit takar az a homályos utalás, hogy „meglehetősen sok funkciót csak POKE utasításokkal lehet elérni". Az interpreterről csak annyit, hogy az ilyen egyszerű és olcsó gépeknél nem teljesen tipikus például a file-kezelésnek az a széles köre, amit a VIC–20 tud (kazettás, disces, RS–232). Egyébként aki az eredeti interpreternél jobbat akar, annak rendelkezésére áll két olyan program (cartridge), amely kibővíti az utasítás készletét. Ez a PROGRAMMER'S AID és a SUPER EXPANDER.

12. kín. Az egész teszthez való hozzáállás és „hozzáértés" tükröződik ebben a pontban. A C 64-gyel teljesen megegyező editor bonyolultnak van titulálva, és állítólag „vér-verítékes küzdelem" kell a gép megismeréséhez. A vallatók okosabbak maradtak volna, ha nem vallatnak.

13. kín. A VIC–20-szal teljesen megegyező BASIC-kel és editálási lehetőségekkel rendelkező C 64-et annak idején a szakemberekből álló vallatók „túlságosan is emberközeli"-nek minősítették. Hát erről ennyit!

+ 1 kín. Az én kínom az, hogy nemcsak ezt az egy pontot értékelték szubjektíven a vallatók, hanem a többi 13-at is! Ha a VIC 20 azokra a funkciókra, amelyek teljesen azonosak a C 64-gyel, ugyanazt az osztályzatot kapná, mint annak idején a C64, akkor az átlaga máris 4,0 lenne, és akkor még az árról nem is beszéltünk!

Összegezve az eddigieket, az én szubjektív véleményem az, hogy *ne a VIC–20-at felejtsük el, hanem a márciusi Vallatót!*

Minden eddigi morgások ellenére maradok a BIT-LET hűséges olvasója.

Szilvágyi Gábor 2092 Budakeszi, Arany J. u. 7. fsz. 3.

Az Ötlet 1985. március 28-i számában megjelent VC 20 vallatóhoz szeretnék hozzászólni. Szerintem több kínnál is hátránynak vették azt, amit a C 64-nél előnyként soroltak fel. (Nem voltam rest, és elővettem a C 64 vallatóját.)

Az alábbiakban felsorolom azokat a kínokat, melyekkel nem értek egyet (bár ez nem sokat jelent).

2. kín: perifériák. Ebben Önök is elismerik, hogy a VC 20-hoz minden csatolható és kapható, csakúgy, mint a C 64-hez. Ebben a két gép megegyezik, mégis a VC–20 8 tizeddel kevesebbet kapott. Miért?

3. kín: képernyőkezelés. Tényleg nem kiváló. Egyébként azt írják, hogy finomgrafika „kizárólag POKE utasításokkal" érhető el. A C 64 alapgépen nem? Különben a kizárólagosságot is megkérdőjelezem, ugyanis a SUPER bővítő modullal (VC 1211A) 160x160-as felbontás érhető el, és egyszerűen kezelhető grafikai utasításokat is tud (pontrajzolás -törlés, körrajzolás, egyenesrajzolás-törlés, kifestés stb.). Az azonban igaz, hogy a C 64 hardver tekintetében kiforrottabb, többet tud. Ha a C 64-nél is az alapgépet osztályozták, hogy kaphatott 4,8-et?

Ha pedig azt bővítéssel vallatták (SIMON'S BASIC, SUPERGRAPHIC), akkor a VC 20-nál miért nem nézték meg a bővítéseket?

5. kín: kazettás tárolás. Ehhez csak annyit, hogy lassú (de kapható hozzá TURBO-modul, ami a 10-szeresére (!) gyorsítja a felvételeket, beolvasásokat), az azonban nem igaz, hogy a gyári magnó mással nem helyettesíthető. Mi két és fél éve SANYO-val használjuk, megbízhatóan. A discet pedig éppúgy kezeli, mint a C 64.

6. kín: gépi kódú programozás. Ennél a kínnál az inkvizítorok azt osztályozták, hogy milyen fordítóprogramok kaphatók a géphez. (Egyébként a VC 20-hoz is lehet jót kapni!) Hogy lehet 1,4 jegy különbség a két processzor gépi kódú programozásában, holott azok szoftver-kompatibilisek? Egy apróság: a C 64 alapgépet mennyivel könnyebb gépi kódban programozni?

7. kín: megbízhatóság. Miben megbízhatatlanabb a C 64-nél?

8. kín: billentyűzet. Ez bosszantott fel a legjobban. A C 64-nél csillagos ötöst akartak adni, a VC 20 4,4-et kapott, pedig a billentyűzetük egy az egyben ugyanaz! UGYANAZ!

10. kín: editálás. Önök is elismerik, hogy editora azonos a C 64-esével. Akkor miért kapott 4 tizeddel kevesebbet? És ki bonyolódhat bele? Ennél egyszerűbb, jobb javítási lehetőség nincs.

11. kín: a gép programnyelve. Hát ez tényleg nem jó! És a C 64 alapgépé? A programnyelvük azonos, mégis a VC 20 1,2-del rosszabbat kapott a C 64-nél! Miért?

12. kín: tanulhatóság. A C 64-nél írják: „...a gép felépítése, kezelése valóban olyan logikus és egyszerű, hogy még megfelelő használati utasítás nélkül is megérthető". A VC 20-ra ez nem igaz? És miért bonyolult a javítási lehetőség? (Erről jut eszembe: ha ez bonyolult, akkor a HT 1080Z javítási lehetősége milyen?)

Az előzőekben felsorolt ellenvetésekkel nem azt akarom mondani, hogy a VC 20 olyan, mint a C 64. Nem, erről szó sincs! De ami a két gépen teljesen egyforma, azt miért nem egyformán osztályozták? Még akkor is, ha az inkvizítorok mások voltak!

Pintér Károly 4. osztályos tanuló 8000 Székesfehérvár József Attila út 59.

A vádak súlyosak. Azonos dolgokat másként osztályoztak az inkvizítorok a C 64-en és a VC 20-on. Nos, mindenre nincs mentség. De az igazsághoz hozzátartozik kedves olvasónk, hogy azonosságok – s ez már a leveléből is kiderül – itt-ott csak majdnem azonosságok. A válasz másik része pedig ott van az Ön levelének utolsó mondatában. Az inkvizítorok nem mindig ugyanazok, ezért az osztályzatok is hol szigorúbbak, hol enyhébbek. Sajnos ez Vallató rovatunk gyengéje. Sajnáljuk, mégis szeretjük ezt a rovatot betegségeivel, korlátaival együtt. De levele alapján úgy vesszük észre, bármilyen mérges most, azért titkon Ön is szereti a Vallatót!

PRIMO



Nagy érdeklődéssel vettem kezembe a BIT-LET március 28-i számát, ugyanis érdekelt, hogy mit írnak a PRIMO-ról. A cikk végigolvasása után azonban egy kicsit dühös voltam a cikk írójára. Ez nem csoda, hiszen olyanokat írt, amik joggal felháboríthatták bármelyik jobbérzésű PRIMO-tulajdonost. Engem konkrétan két dolog bántott.
Az egyik a kazettás tárolással kapcsolatban írtak volt. Nem tudom, hogy milyen „kaliberű" magnókkal rendelkezik a tisztelt író, de nekem egy MK 29-es magnó nagyon jól működik. Ez, azt hiszem, eléggé kis kaliberű. Sem a felvételnél, sem a lejátszásnál nem volt még probléma. Még a DEMO kazettánál sem. Egyébként a DEMO kazettán meg vannak jelölve az oldalak, és valaki könnyen megjegyezheti, hogy a B oldal a hibás oldal.
A másik dolog a billentyűzet. Abban egyetértek Szőke Ferenccel, hogy a PRIMO billentyűzete nem a legkedvezőbb, azonban megfelelő beállítás esetén nem okoz gondot a gépelés (nálam legalábbis). A játékprogramoknál tényleg gondot okoz a billentyűzet, azonban gondoljunk csak arra, hogy a számítógépet elsősorban nem játékra kell használni. A dokumentációval kapcsolatban nekem is az a véleményem, amit a cikkben olvashattunk. A jegyet azonban egy kicsit erősnek tartom. A gép szerintem bőven megüti a 3-as szintet, ha ennél nem is jobb sokkal. Egy megjegyzés. A PRIMO-hoz közölt (illetve nem közölt) programoknak a túl kevés számát jó lenne gyarapítani.
És végül, hogy ne csak kritizáljak – egy észrevételemet szeretném közkinccsé tenni. Próbáljuk ki a következő programot:

```
10 CLS
20 FOR I=1 TO 2
30 POKE 16457,1
40 ?"PRIMO PERSONAL COMPUTER" (Ide bármilyen szöveget lehet írni)
50 NEXT
```

Meglepve tapasztalhatjuk, hogy a képernyő aktuális helyére kiírt szöveg karakterei megvastagodnak. Észrevehetjük azonban azt, hogy a szöveget nem lehet PRINT $-ral kiírátni. Csak úgy, hogy a szöveg elé megfelelő számú PRINT-et írunk, majd ha elértük az aktuális sort, ott PRINT TAB(x) "szöveg"-gel kiíratjuk a képernyőre az üzenetet.
Próbáljuk ki a programot úgy, hogy a ciklusváltozó második értéke ne 3 legyen, hanem annál nagyobb. Valamint úgy, hogy a ciklus 180–210-ig terjedjen. Ezt a programot ki lehet egészíteni különböző vezérlő karakterekkel, az eredmény azonban nem minden esetben változik. Remélem, levelem nem talál süket fülekre, és nem kerül a szemeteskosár mélyére!
Nógrádi Előd (2. oszt. gimn. tanuló, Szeged)

Múlt havi számunkban tettük közzé fölhívásunkat a „szoftver ötletek" című új rovat indításáról. Máris érkeztek anyagok, amelyek eleget tesznek kérésünknek, miszerint egy-egy programozási ötletet adnánk közre ebben a rovatban.

RND(X)

Egy olvasó felvetette a márciusi BIT-LET-ben, hogy hogyan lehet az iskolaszámítógép véletlenszám generátorának kezdőértékét beállítani. Egyes esetekben valóban szükség lehet arra, hogy a program ismételten ugyanazt a véletlenszám sorozatot állítsa elő.
Az iskolaszámítógép a véletlenszámok előállításához a 16554–16556 bájtok tartalmát használja fel (RND-mag). Ha ezekre a címekre ugyanazokat az értékeket tároljuk le, akkor a véletlenszám sorozatok is megegyeznek. A POKE 16554,11:POKE 16555,11:POKE 16556,11 utasítások után pl. az RND(100) a következő sorozatot állítja elő: 2, 67, 17, 70, 15, 79, 33, ...
A régebbi típusú (ékezetes betűket nem ismerő) számítógépeknél bekapcsolás után ezeknek a bájtoknak a tartalma tapasztalataim szerint rendre 255, 0, 255, így ezek a gépek ki-bekapcsolás után külön beállítás nélkül is ugyanazt a sorozatot hozzák létre. Az újabb típusú gépeknél viszont a bekapcsolás után is változó az RND magja. A RANDOM utasítás különben a 16555 bájt tartalmát változtatja meg véletlenszerűen (az R regiszter tartalmát másolja át).

SZOFTVER ÖTLETEK



Gépi kódú programban is szükség lehet véletlenszámok előállítására Az RND(X)-et előállító szubrutin a 14c9H címen kezdődik. A behívás előtt az X paramétert az ún. szoftverakkumulátorba kell letárolni; egész típusú szám esetén a 4121–4122H bájtokba, egyszeres pontosságú számnál a 4121–4124H bájtokba, a gép belső számábrázolásának megfelelően. A 40AFH bájtba egész számnál 2-t, egyszeres pontosságúnál 4-et kell tölteni. A véletlenszám ugyanezekbe a bájtokba kerül, egyszeres pontosságú alakban.
Némileg gyorsabban működik a rutin, ha az X paramétert a HL regiszterpárba töltjük, s a rutint a 14D5H címen hívjuk meg. Ekkor azonban az X csak pozitív kettes komplemens egész szám lehet. X=0 esetén pedig a 14FOH kezdőcímen hívhatjuk meg a rutint, s ez nem igényel semmilyen előkészítést. A véletlenszám ezekben az esetekben is a szoftver-akkumulátorba kerül.
A következő programrészlet pl. a képernyőre rajzol ki véletlenszerűen 191-es kódú karaktereket:

```
      LD    B,100        ; A ciklus előkészítése
CIK:  PUSH  BC
      LD    HL,1024      ; A paraméter letárolása
      LD    (4121H),HL
      LD    A,2          ; Az egész típus beállítása
      LD    (40AFH),A
      CALL  14C9H        ; Véletlenszám generálás
      CALL  0B26H        ; Konvertálás egészre
      LD    HL,(4121H)   ; Egy képernyő-memóriacím előállítása
      LD    BC,3C00H
      ADD   HL,BC
      DEC   HL
      LD    (HL),191     ; Karakter kivitele a képernyőre
      POP   BC           ; Ismétlés 100-szor
      DJNZ  CIK
```

A RANDOM rutin kezdőcíme 01D3H, de ez csak a következő három utasítást tartalmazza:

```
LD   A,R
LD   (40ABH),A
RET
```

Juhász Tibor tanár, Zalaegerszeg

```
1 REM DARDAY ZOLTAN
2 BORDER 7: PAPER 7: INK 1
5 GO SUB 800: LET n=n-450
7 CLEAR n
8 GO SUB 800: LET n=n+1
9 LET s=n: GO SUB 2000
10 FOR i=n TO n+407: READ a: POKE i,a: NEXT i
200 CLS : FOR i=0 TO 21: PRINT "TEGLALAPSCROLL,JO
BBRABALRA,LEFEL": NEXT i
210 IF INKEY$="7" THEN  RANDOMIZE USR n
220 IF INKEY$="8" THEN  RANDOMIZE USR (n+131)
230 IF INKEY$="5" THEN  RANDOMIZE USR (n+204)
240 IF INKEY$="6" THEN  RANDOMIZE USR (n+277)
245 IF INKEY$="1" THEN  GO TO 5000
250 GO TO 210
800 LET n=PEEK 23731*256+PEEK 23730
805 PRINT "RAMTOP:";n
810 RETURN
1000 DATA 221,42,178,92,33,0,64,62,3,8,221,35,221,
126,0,221,35,221,110,0,254,0,32,11,8,61,40,47,6,8,
36,16,-3,24,-26,6,7,197,93,84,36,237,75,11,92,237,
176,237,75,10,92,43,16,-3,193,16,-20
1010 DATA 93,84,6,32,35,16,-3,61,32,44,237,83,176,
92,8,61,32,13,237,91,176,92,237,75,10,92,18,19,16,
-4,201,237,75,11,92,245,62,32,145,237,176,254,0,40
,4,71,19,16,-3,241,98,107,24,-102
1020 DATA 6,7,37,16,-3,237,75,11,92,237,176,237,75
,10,92,43,16,-3,24,-96
1030 DATA 221,42,178,92,33,0,64,62,3,8,221,35,221,
126,0,221,35,221,110,0,254,0,32,10,6,8,36,16,-3,8,
61,200,24,-25,6,8,197,229,245,183,237,75,10,92,245
,62,32,144,203,30,35,16,-5,254,0,40,4,71,35,16,-3,
241,61,32,-26,241,225,36,193,16,-35,24,-44
1040 DATA 221,42,178,92,33,255,87,62,3,8,221,35,22
1,126,6,221,35,221,110,6,254,0,32,10,6,8,37,16,-3,
8,61,200,24,-25,6,8,197,229,245,183,237,75,10,92,2
45,62,32,144,203,22,43,16,-5,254,0,40,4,71,43,16,-
3,241,61,32,-26,241,225,37,193,16,-35,24,-44
1050 DATA 221,42,178,92,33,255,87,62,3,8,221,35,22
1,126,6,221,35,221,110,6,254,0,32,11,8,61,40,47,6,
8,37,16,-3,24,-26,6,7,197,93,84,37,237,75,11,92,23
7,184,237,75,10,92,35,16,-3,193,16,-20
1060 DATA 93,84,6,32,43,16,-3,61,32,44,237,83,176,
92,8,61,32,13,237,91,176,92,237,75,10,92,18,27,16,
-4,201,237,75,11,92,245,62,32,145,237,184,254,0,40
,4,71,27,16,-3,241,98,107,24,-102
1070 DATA 6,7,36,16,-3,237,75,11,92,237,184,237,75
,10,92,35,16,-3,24,-96
2000 PRINT AT 4,7;"SOR"
2001 INPUT "honnan?",t
2002 LET t=INT t: IF t<0 OR t>23 THEN  GO TO 2001
2003 INPUT "meddig?",g
2004 LET g=INT g: IF g<t OR g>23 THEN  GO TO 2003
2005 PRINT t,g;AT 11,7;"OSZLOP"
2007 INPUT "honnan?",k
2008 LET k=INT k: IF k<0 OR k>31 THEN  GO TO 2007
2009 INPUT "meddig?",v
2010 LET v=INT v: IF v<k OR v>31 THEN  GO TO 2009
2012 POKE 23563,v-k+1
2015 PRINT k,v
2026 FOR b=1 TO 2
2028 FOR i=1 TO 3
2030 LET a=(t<(i*8) AND t>=((i-1)*8))*(i*8-t)
2031 LET a=a+(t<((i-1))*8 AND g>=((i-1)*8))*8
2032 LET a=a+(g<(i*8) AND g>=((i-1)*8))*(g-i*8+1)
2035 POKE n,a: LET n=n+1
2040 LET f=(t<(i*8) AND t>=((i-1)*8))*32*(t-8*INT
(t/8))
2042 IF b=2 THEN  LET f=224-f+v
2043 IF b=1 THEN  LET f=f+k
2045 POKE n,f: LET n=n+1
2060 NEXT i
2065 LET i=23-t: LET t=23-g: LET g=i
2070 NEXT b
2075 PRINT AT 18,10;"PRESS"
2080 PAUSE 0
2100 RETURN
5000 CLS : PRINT "Oszlop: ";PEEK 23563
5050 SAVE "SCROLLcode"CODE s,450
5090 STOP
9000 SAVE "*SCROLL..*" LINE 1
```

PROGRAM AJÁNLAT

SCROLL Spectrum

Az alábbi program a képernyő előre meghatározott részét tudja 4 részben scrollozni. Ehhez hasonlót (teljes képernyő, jobbra-balra) már közöltünk, de megítélésünk szerint ez is számot tarthat a közlésre.

A program be is mutatja a gépi kódú rutin működését, azaz az 5, 6, 7, 8, billentyűk megnyomásásra értelemszerűen egy pixellel a megfelelő irányba mozdítja az ábra kijelölt részét.

A gépi kódú programot megkapjuk kazettán, ha az 1-es billentyűt megnyomjuk.

A gépi kódú programot az 5000–5090 sorok viszik ki a kazettára.

„SCROLLcode" felhasználása MÁSIK programban, pl.

10 CLEAR X
20 LOAD""CODE(X+1),450 betöltés, inicializálás
30 POKE 23563, OSZLOP

900 IF INKEY$="7" THEN RANDOMIZE USR (X+13)
910 IF INKEY$="8" THEN RANDOMIZE USR (X+144)
920 IF INKEY$="5" THEN RANDOMIZE USR (X+217)
930 IF INKEY$="6" THEN RANDOMIZE USR (X+290)

Az előző programban X=konstans érték, így például
10 CLEAR 29999
920 IF INKEY$=„5" THEN RANDOMIZE USR 30216
OSZLOP szintén konstans érték (1–32), értékét az x SCROLL...X program 5000 sora írja ki a „SCROLLcode" gépi kódú program kazettára vitele előtt.

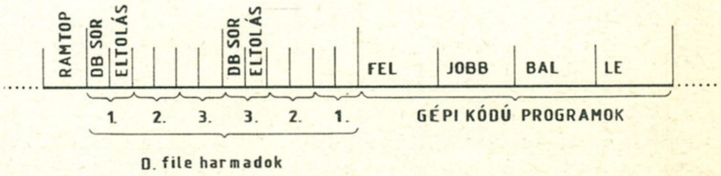
Egyéb tudnivalók

A gépi kódú program a 23563–4 DEFADD rendszerváltozót is használja, amelyek értéke a program során megváltozhat.
BREAK hatására 23563 értéke 0 lesz, így a gép „megbolondul:".
DEFADD rendszerváltozó értéke FN függvényhívás hatására is megváltozik.

A hiba kiküszöbölése:
BREAK ill. FN után POKE 23563, OSZLOP

x SCROLL...x programban használt billentyűk:
5, 6, 7, 8 gépi kódú programot aktivizálják
1 gépi kódú program kazettára vétele

„SCROLLcode" elhelyezkedése a tárban



A gépi kódú program assembler listáját nem közöljük, mert túlzottan hosszú.

Dárday Zoltán

ötlet

KEZDŐKNEK HT HALADÓKNAK grafikáról!

Az iskolaszámítógép grafikájával kapcsolatban már több cikk is megjelent a BIT-LET-ben [lásd Halász Péter cikkét a 11. (1984. VIII. 28.) Török Turul cikkét a 15. (1984. XII. 18.) számunkban], mégis (főleg a HT GÉPNYERŐ kiértékelése közben szerzett tapasztalatokra támaszkodva) úgy érzem, hogy nagyon sokan nem ismerik vagy nem tudják jól használni azt az alapvető módszert, melynek segítségével BASIC-ben is elég gyors grafikát használhatunk. Ezért most főleg erről lesz szó.

Mint tudjuk, az iskolaszámítógépen nincs a szó igazi értelmében vett grafika, csak úgynevezett grafikus karakterek vannak (ezek ismeretét ezentúl feltételezzük; lásd például az Ismerd meg a Basic nyelvjárásait! című könyvben). A képernyőre rajzolni négyféleképpen lehet: a SET utasítással, a POKE utasítással, gépi kódú rutinnal, vagy a grafikus karakterek közvetlen kiírásával. Az első két módszer az esetek többségében nagyon lassú, a harmadikat meg csak azok tudják használni, akik tudnak gépi kódban programozni, s sokszor még nekik sem érdemes ezt a módszert választani. Ezért a 4. módszerről szeretnék leírni néhány dolgot.

A módszer azon alapszik, hogy egy stringben tároljuk a megfelelő grafikus (és egyéb) karaktereket, s utána a stringet egy PRINT@ utasítással íratjuk ki a képernyő megfelelő helyére. A PRINT @ I, A$ utasítás hatására a kurzor az I. pozícióra áll (ha 0 ≤ I ≤ 1023, egész), azaz a képernyő INT(I/64)+1. sorának I – 64 * INT (I/64) + 1. pozíciójára, s utána erre a helyre írja ki az A$ első karakterét, majd a kurzor az eggyel magasabb sorszámú helyre ugrik, ezután, ha van a stringnek 2. karaktere, azt írja ki, stb. Azonban a kurzornak ezt az egyesével lépegető mozgását megváltoztathatjuk, ha a stringben vezérlő karaktereket helyezünk el. A fontosabb vezérlő karakterek a következők:

ASCII kód	hatás
8	a kurzor 1 karakterrel visszalép (általában az utoljára kiírt karakter helyére), s az ott levő karaktert kitörli
10	soremelés
13	NEW LINE
24	a kurzor 1 karakterrel visszalép (de nem töröl)
25	a kurzor 1 karakterrel előbbre lép
26	a kurzor egy sorral lejjebb lép (a sor ugyanannyadik pozíciójába, ahányadikban volt)
27	a kurzor egy sorral feljebb lép
28	a kurzor a bal felső sarokba lép
29	a kurzor az aktuális sor elejére lép
32	betűköz – a kurzor törli a pillanatnyi helyén levő karaktert, majd eggyel jobbra lép
127	pepita karakter
192–255	kód – 192 db betűköz, azaz ennyit lép a kurzor jobbra, s közben töröl.

Ezek a karakterek például a CHR$ függvény segítségével a string megfelelő részein elhelyezhetők [a CHR$ (220) csak 1 db karakter, melynek kiíratáskor a hatása 28 db betűköz kiírásával egyezik meg]. Megjegyezzük még, hogy mivel egy string csak 255 karakter hosszú lehet, így előfordul, hogy ábránk nem fér el egy stringben. Ez nem nagy baj, folytathatjuk egy másik stringben, lehetőleg úgy, hogy ezt az előző után még ugyanabban a PRINT utasításban ;-vel elválasztva lehessen kíratni.

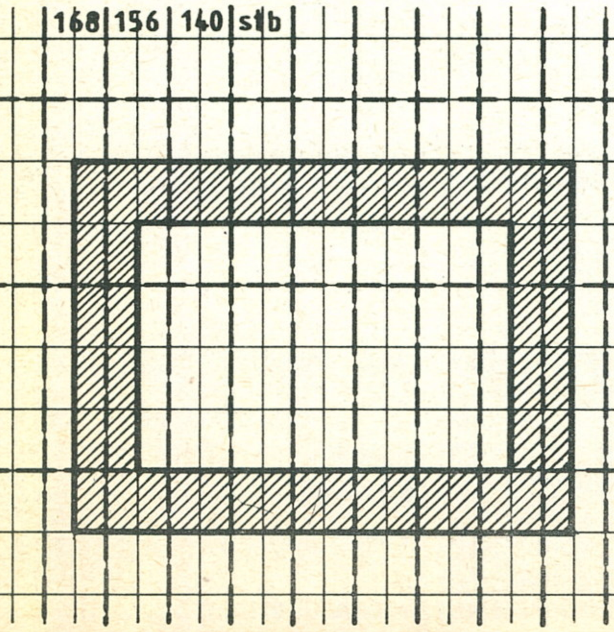
Hogyan kódoljuk a megrajzolandó ábránkat?
Az egyik megoldás az, hogy írunk erre a célra egy segédprogramot. A másik az, hogy „kézzel" kódoljuk az ábrát. Ehhez ajánlatos franciakockás papírt és egy kódtáblázatot használni (melyet vagy magunk készíthetünk, vagy használjuk valamelyik könyv grafikus kódokat is tartalmazó táblázatát). Az ábrát lerajzoljuk a franciakockás papírra (azért ilyenre, mert ennek téglalapjai hasonlítanak legjobban a gép által megjelenített pontokra), majd beosztjuk a karaktereknek megfelelően 3x2-es nagyobb téglalapokra. Ezután a táblázat segítségével és egy kis gyakorlattal könnyen elvégezhető a kódolás, közbeiktatva a megfelelő vezérlő karaktereket. Megjegyezzük, hogy az ábrának nem kell feltétlenül a bal felső sarokban kezdődni, hiszen a string kezdődhet vezérlő karakterekkel is; ennek néha praktikus előnye van – lásd például az első példát.

Hogyan tároljuk a programban az ábrát? Természetesen egy (vagy több) stringben, de ezeknek a stringeknek értéket is kell adni. Az ábrát a megfelelő stringbe a következő módszerek segítségével „vihetjük" be:

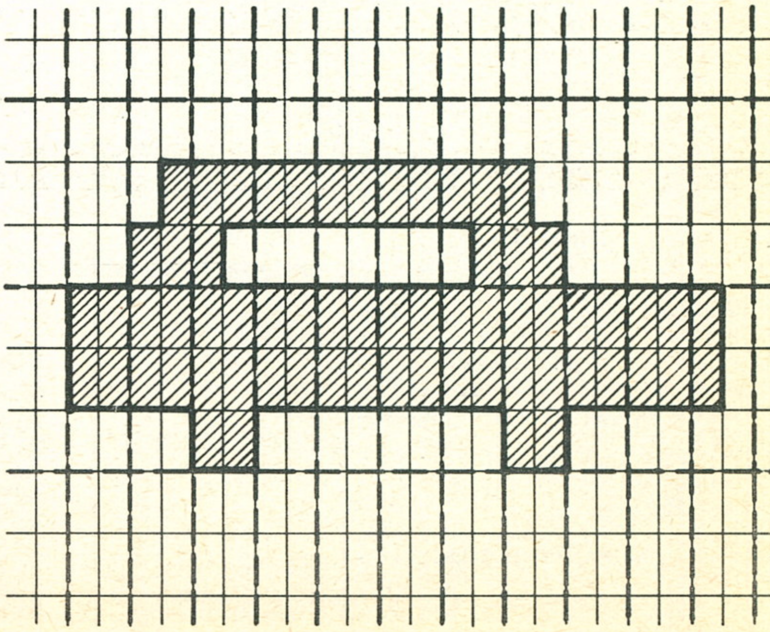
1. Egyszerű értékadással. Használjuk a STRING$ és a CHR$ függvényeket!

2. A megfelelő kódokat felsoroljuk egy DATA utasításban.

3. Ezt finomíthatjuk úgy, hogy ha egy kód egymás után többször fordulna elő, akkor



1. ÁBRA



2. ÁBRA

például elé írjuk mínusz előjellel azt, hogy hány darab kell belőle, s utána egyszer leírjuk a kódot. (Lásd az 1. és 2. példákat!) Ha egész hosszú részek is ismétlődnek, megfelelő kódok beiktatásával ezeket is kezelhetjük az előbb leírt módon vagy a következő móddszerrel:

4. Kombinált módszer. Ha az ábrában több azonos részlet van, először ezeket valamelyik módszerrel segéd-stringekbe rakjuk, majd használjuk a stringek összefűzésére (+) műveletet.

5. A stringet közvetlenül tároljuk a DATAban. Erre nézve lásd Halász Péter említett cikkét. Ezt a módszert csak akkor érdemes használni, ha valamilyen oknál fogva az is fontos, hogy a futás közben a stringek feltöltésének ideje is rövid legyen – ez elég ritkán fordul elő.

Példák, feladatok

1. Néha előfordul, hogy a képernyőn néhány dolgot be szeretnénk keretezni. Erre mutat példát az 1. program. A keretet úgy kódoljuk, hogy tetszőleges hosszúra meg lehessen csinálni. A K változó értéke mutatja, hogy milyen hosszú szöveget akarunk írni a keretbe, s az adatsorban „-i"-vel azt fogjuk jelölni, hogy a következő karakterből K+i–1 db-ot akarunk bevenni a stringbe. Az ábrát úgy készítjük, hogy a beírandó szöveg első karakterpozícióján kezdődjön, úgy nem kell új kiírási helyet számolnunk. A keret közepét betűközök helyett jobbra mozgatásokkal töltjük fel, így a már kiírt szövegre is rá lehet íratni a keretet, nem törli ki (l. 1 ábra).

1. A kódolt ábrát a program 900-as sorában találhatjuk meg. A 900–1020 sorokig terjedő rutin végzi el a K hosszúságú keret előállítását, a program többi része csak egy bemutató program, mely a szubrutin használatát és a megjelenítést mutatja be egy példán, mely a HT GÉPNYERŐ 1. feladatában lett volna felhasználható: egy számológép billentyűit és kijelzőjét rajzolja a képernyőre.

2. Ebben a példában egy egyszerű mozgást illusztrálunk, egy kocsi szalad a képernyő bal szélétől a jobb széléig. A kocsit a 3-as módszerrel rakjuk a B$-ba, az utat az 1-essel a C$-ba. A rutin majdnem ugyanaz, mint az előbb, csak nincs szükség a K változóra, s B$ tömb helyett egyszerű B$-t használhatunk. Érdemes kipróbálni 100-as sebességgel, valóban imponáló gyorsasággal fog a kocsi végigszaladni a képernyőn (l. 2. ábra).

A kódolásban vegyük észre, hogy az ábrához „hozzáragasztottunk" a kocsi hátuljánál 1–1 betűközt, ez biztosítja, hogy az előző állapotban lévő kocsi végét letöröljük. Ezt a megoldást minden mozgatásnál érdemes alkalmazni. A program tovább gyorsítható, ha kivesszük a lassító ciklust (mely a sebességet szabályozza), s még jobban, ha a kocsit csak minden második pozícióra rajzoljuk ki, persze ekkor új kódolást kell csinálnunk, a kocsi végéhez 2–2 betűközt kell ragasztani. Figyeljük még meg, hogy a mozgó kocsi és az álló út „találkozása" pont karakterhatáron van, lehetőség szerint hasonló esetekben ezt mindig alkalmazni kell!

3. Feladat: készítsünk egy nagyobb ábrát, mely nem végez haladó mozgást, de egyes részei mozognak (pl. egy tornázó ember). Az ábra álló és az egyes mozgó részeit külön stringekben érdemes tárolni (illetve a fázisokat string-tömbökben), s itt is vigyázni kell az előző fázis törlésére.

4. Feladat: most már olyan ábrát is készíthetünk, melynek egyes részei is mozognak, s az egész haladó mozgást végez (pl. egy nagy madár repül a bal alsó sarokból a jobb felsőbe).

Aki a cikket figyelmesen elolvasta, a példákat megértette (esetleg ki is próbálta), és a feladatokat megoldotta, az bátran hozzáfoghat bármilyen komolyabb grafikát használó program írásához. Jó kísérletezést!

S még egy apró tanács, mely nem a grafikára vonatkozik. Biztos sokan jártak már úgy, hogy hosszabb program beolvasása során a gép 1–2 karaktert vett csak be rosszul, ez a futás során esetleg csak később derül ki, vagy rejtve is maradhat. Különösen kényelmetlen ez a programok másolásakor. A megoldás egyszerű: a bevitel után a szalagot újra a program elejére állítjuk, s CLOAD?-lel összehasonlítjuk a kazettán lévő és a gépben lévő programot. Ha ez a próba sikeres, akkor (majdnem) biztosak lehetünk abban, hogy a gépben pontosan a kazettán tárolt program van. Hogy az jó-e? Azt mi sem tudhatjuk ...

Király Zoltán

1. PROGRAM

```
2 CLEAR 1000
5 CLS
10 DIM A$(30),B$(40)
20 FOR I=1 TO 26:READ Q$:NEXT I:L=0
30 L=L+1:READ A$(L):IF A$(L)<>"" THEN 30
32 L=L-1:FOR I=1 TO L:K=LEN(A$(I)):IF B$(K)="" THEN GOSUB 1000
33 NEXT I
35 K=38:GOSUB 1000:PRINT @79,B$(K);
40 FOR I=0 TO (L-1)/5:FOR J=1 TO 5
50 K=LEN(A$(I*5+J))
60 P=I*3*64+J*9+326:PRINT @P,A$(I*5+J);:PRINT @P,B$(K);:NEXT J,I
70 IF INKEY$="" THEN 70 ELSE END
900 DATA 24,24,27,168,156,-1,140,172,148,26,-5,24,170,149,-1,25,170,149,26,-5,24
,130,-3,131,129,0
1000 B$(K)="":KK=1:RESTORE
1010 READ W:IF W=0 THEN RETURN ELSE IF W<0 THEN KK=K-W-1:GOTO 1010
1020 B$(K)=B$(K)+STRING$(KK,W):KK=1:GOTO 1010
1500 DATA X[2,7,8,9,/,STO,4,5,6,*,RCL,1,2,3,-,EXC,0,.,+/-,+,""
```

2. PROGRAM

```
2 CLEAR 1000
5 CLS
10 GOSUB1000:C$=STRING$(64,143)
20 INPUT "SEBESS@G (1-100)";V:IF V<1 THEN 20
30 CLS:PRINT @640,C$;
40 FOR I=0 TO 52:PRINT @512+I,B$;:FOR J=1 TO 100/V:NEXT J,I
50 IF INKEY$="" THEN 50 ELSE PRINT:PRINT:PRINT:PRINT:END
900 DATA 32,32,184,156,-3,140,172,180,-9,24,26,32,143,143,191,-4,143,191,143,143
,133,0
1000 B$="":KK=1:RESTORE
1010 READ W:IF W=0 THEN RETURN ELSE IF W<0 THEN KK=-W:GOTO 1010
1020 B$=B$+STRING$(KK,W):KK=1:GOTO 1010
```

A RAM

ÉS A KÉPERNYŐ

A személyi számítógépeknél a memória jelentős részét a gép
fenntartja egy viszonylag egyszerű feladatra: a tv-képernyőjén
megjelenő kép adatainak tárolására és feldolgozására. Így van
ez a Spectrumnál is, és erről a gép kézikönyve eléggé szűk-
szavúan beszél. A lényeget természetesen közli: a RAM
elején a 16384 címtől kezdve a „Display file" majd a 22528-cal
kezdődően az „Attributes file" helyezkedik el.
A részletesebb megtárgyalást érdemesebb a másodikkal
kezdeni, mert annak egyszerűbb a felosztása és működése.
Az attribútumok (jellemzők) az egyes karakterek jellemzői-
nek meghatározásával az egész kép milyenségét alakítják ki.
Ezek jól ismertek: FLASH, BRIGHT valamint a PAPER és
INK színek.
Az „Attributes file" 24*32 = 768 byte-ot foglal el. Érdemes
a kezdő és befejező címeket hexadecimálisan is felírni:

	dec.	hex.
Kezdőcím	22528	5800
Utolsó cím	23295	5AFF

A fenti 768 byte-ból az első 22*32=704 a PRINT (és PRINT AT) területe míg a fennmaradó 2*32=64 byte a hibaüzenetek és egyéb célokra van fenntartva.
Ezt figyelembe véve a PRINT terület utolsó címe: 23231 (dec.), illetve 5ABF (hex.).
Közismert, hogy a sorokat 0-tól 21-ig számozzuk (a 22. és 23. sor az említett fenntartott sor), az oszlopokat pedig 0-tól 31-ig.
Ha a sor száma = „s" és az oszlop száma = „k", akkor bár-mely karakter attribútum decimális címe:
22528+32*s+k
Ezeken a címeken egy-egy byte (8 bit) kerül tárolásra a következők szerint:

FLASH	BRIGHT	PAPER	INK
1 bit	1 bit	3 bit	3 bit
0–1	0–1	0–7	0–7

Ezt legjobb egy példával illusztrálni, legyen a példa byte-ja (kettes számrendszerben): 11010101

FLASH	BRIGHT	PAPER	INK
1	1	0 1 0	1 0 1
be	be	piros	cyan

Vegyük a példához a 4. sor 5. oszlopát (karakterhelyét).
Ez biztosan ismert minden Spectrumot használó számára: a PRINT AT 4, 5-nek felel meg.
Az attribútum címe : 22528 + 32*4 + 5 = 22661

Itt a példában szereplő byte decimális alakját szokásos használni: az 11010101 megfelel 213 (dec.)-nek.
Így a közvetlen parancs:

POKE 22661,213

Természetesen így az INK szín nem látható, csak a PRINT AT 4,5; "x" után.

Ha a képernyő egyes pozícióinak címét hexadecimálisan írjuk fel, a következőt kapjuk:

oszlop	0.	1.	2.		29.	30.	31	
0. sor	5800	5801	5802	.....	581D	581E	581F	PRINT terület
1. sor	5820	5821	5822	.....	583D	583E	583F	
2. sor	5840	5841	5842	.....	585D	585E	585F	
.							.	
.								
.								
21. sor	5AA0	5AA1	5AA2	.....	5ABD	5ABE	5ABF	
22. sor	5AC0	5AC1	5AC2	.....	5ADD	5ADE	5ADF ←	hibaüzenet sorok
23. sor	5AE0	5AE2	5AE2	.....	5AFD	5AFE	5AFF ←	

Így már jól látható, hogy miért célszerű hexadecimális számokkal felírni a képernyő attributum (jellemzők) címeit.
Az egyes pozíciók számértéke jól mutatja a helyet is meg a címet is.
A „Display file" egy kissé bonyolultabb, mint az „Attributes file", mert minden karakter 8 byte segítségével írható le.
Ennek részletezése nem szükséges, hiszen a Spectrum használói az UDG karakterek beírásánál használják a 8×8 pontból álló tömböt.

Amikor egy program betöltése a LOAD SCREEN $-ral történik, bizonyára mindenki megfigyelte, hogy a kép kirajzolása nem egyszerre történik.

A RAM és a képernyő

Írjuk be az alábbi rövid programot:

```
10 FOR i=0 TO 21
20 PRINT " (32 fekete négyzet: graphics és caps shift 8-cal)"
30 NEXT i
```

A RUN parancs után a 22 sornak megfelelő terület fekete lesz. Ekkor írjuk be : SAVE "akármi" SCREEN $, és rögzítsük magnetofonszalagon. Töröljük a programot és LOAD "akármi" SCREEN $ után töltsük be. Jól látható, hogy a beírás közben kirajzolódik a fekete terület, de nem egyszerre. Először a felső harmad első sorai, aztán a második sorok kirajzolása történik meg, végül a nyolcadik sor, és csak ezután kezdődik a képernyő második harmadának kirajzolása. Itt is először az első sor, aztán a második, majd a harmadik, végül a nyolcadik sor, amivel a képterület második harmada is fekete lesz. Teljesen azonos módon rajzolódik ki a harmadik harmad is.

Így jól látható, hogy a „Display file" három részre van osztva; ezeket a részeket nevezi BLOCK-nak az irodalom.

Írjuk ismét fel a hexadecimális címeket:

	Kezdő cím	Utolsó cím	Képernyősor
1. Block	4000	47FF	0–7
2. Block	4800	4FFF	8–15
3. Block	5000	57FF	16–23

A fenti táblázatból látható, hogy a 3. Block tartalmazza a 22. és 23. „hibaüzenet" sorokat is.

A „Display file" által igényelt byte-ok száma:

32∗24∗8 = 6144 (dec.) megfelel 1800 (hex.)-nek.

Egy-egy Block ennek a harmadát tartalmazza:

6144 / 3 = 2048 byte megfelel 800 (hex.)-nek.

Elegendő az 1. Blockot vizsgálni, mert a másik kettő szerkezete, felépítése teljesen azonos.

Jól ismert a PLOT utasítás 256 x 176 pontból álló koordinátahálózata. Ennek 175. pontsora megegyezik az 1. Block első pontsorával. Ha ebben a rendszerben vizsgáljuk az 1. Blockot, akkor az elsőként kirajzolt pontsorok rendre a következők:

175 , 167 , 159 , 151 , 143 , 135 , 127 , 119

a másodikként rajzolt pontsorok:

174 , 166 , 158 , 150 , 142 , 134 , 126 , 118

harmadszorra:
173 , 165 , 157 , 149 , 141 , 133 , 125 , 117

és végül nyolcadikként:

168 , 160 , 152 , 144 , 136 , 128 , 120 , 112.

Ezek szerint a „Display file" első 32 byte-ja hordozza a 175. pontsor adatait. A 33. byte a 167. pontsor bal szélének értékét mutatja. Győződjünk meg erről a következőképpen:

írjuk be közvetlen parancsként:

POKE 16384, BIN 10101010 vagy ezzel

egyenértékű POKE 16384,170

Erre a bal felső sarokban egy rövid pontsor jelenik meg,

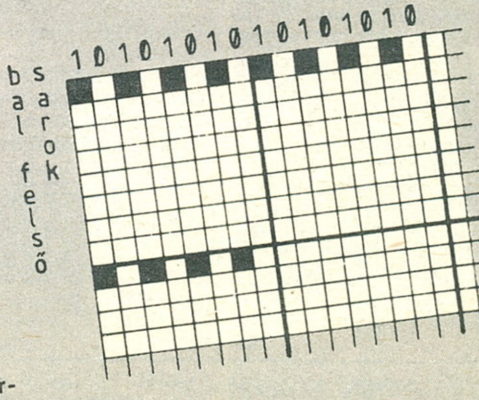
aztán POKE 16385,170

ez az előző folytatásaként rajzolódik ki.

Most növeljük 32-vel a kezdőcímet: 16384 + 32 = 16416

evvel POKE 16416,170 és

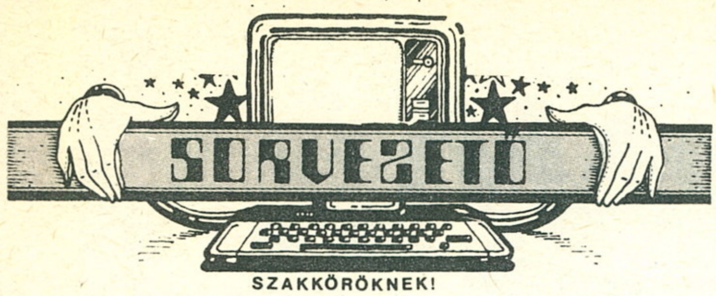
most a rövid pontsor az elsőként kirajzolt alatt a 167. sorban fog megjelenni.



A további pontsorok kirajzolása az előzőkben leírtak szerint folytatódik. A 112. sor kirajzolásával az 1. Block teljesen be fejeződik. Ezután a következő 2048 (dec.) byte kerül sorr teljesen azonos módon mint az az 1. Blocknál történt.

A „Display file" kezelése egy kissé bonyolultabb, mint jellemzőké, de ha tekintetbe vesszük, hogy egy ponts 32 byte-ot igényel és hexadecimálisan számolunk, a mun könnyebbé válik.

Nagy Károly okl. g



ISMÉTLÉS A TUDÁS ANYJA?

Tudnivalók a HT-gép billentyűzetéről

Közismert, hogy a HT–1080 Z gépeknél a bekapcsolás után nem lehet a billentyűzetről kisbetűket bevinni, és az automatikus billentyűismétlés sem működik. Ha e funkciókat használni szeretnénk, akkor a SYSTEM parancs után háromféle lehetőség adódik:

/12288	esetén	kis- és nagybetű villogó kurzor billentyűismétlés
/12299	esetén	kis- és nagybetű billentyűismétlés
/12294	esetén	kis- és nagybetű

A fenti bővítések természetesen behívhatók gépi kódú rutinként is, meghívásuk azonban a BASIC program futásának leállását eredményezi. Így viszonylag nehezen érhető el, hogy a felhasználó, a gép előzetes állapotától függetlenül, a programozó által beállított módon működő billentyűzetet használjon. Ugyanakkor egy-egy alkalmazásnál fontos lehet a kisbetűk használata, valamint az, hogy az ismétlést le is lehessen tiltani. Bekapcsolt billentyűismétlés esetén ugyanis bármely billentyű leütése lényegesen lelassítja a program futását. (Érdemes kipróbálni!) A következőkben e gondok megoldásához szeretnék segítséget adni.

A nagybetű/kisbetű átkapcsolást a 16414–16415 (401EH–401FH) memóriák átírásával végezhetjük, az alábbi táblázat felhasználásával.

Cím		Tartalom		Hatása
16414	401EH	88	58H	Csak nagybetű
16415	401FH	4	04H	
16414	401EH	71	47H	Kis- és nagybetű
16415	401FH	49	31H	

A memóriák tartalma legegyszerűbben a POKE utasítás segítségével módosítható. Parancsként vagy programban, utasításként is kiadhatjuk, a program futása most nem áll le. Nehezebb dolgunk van a billentyűismétléssel. Az átkapcsolás most is két byte átírásával történhet:

Cím		Tartalom				Ismétlés
		régi gép		új gép		
16406	4016H	227	E3H	227	±E3H	nincs
16407	4017H	3	03H	3	03H	
16406	4016H	120	78H	58	3AH	van
16407	4017H	48	30H	48	30H	

Sajnos most a POKE utasítás nem vezet eredményre, a gép teljesen „megvadul" tőle, és csak kikapcsolással bírható normális működésre. Nem módosíthatók a fenti memóriák a monitorprogrammal sem. Egy gépi kódban írt rövid programmal azonban a két cím tartalma átírható, és ezzel a megfelelő billentyűfunkció meghívható. A gépi kódú rutint a 32000–32013 (7D00H–7D0DH) memóriaterületre töltjük.

Cím		Tartalom		Mnemonik
32000	7D00H	33	21H	LD HL,303AH
32001	7D01H	58	3AH	
32002	7D02H	48	30H	
32003	7D03H	34	22H	LD (4016H),HL
32004	7D04H	22	16H	
32005	7D05H	64	40H	
32006	7D06H	201	C9H	RET
32007	7D07H	33	21H	LD HL,03E3H
32008	7D08H	227	E3H	
32009	7D09H	3	03H	
32010	7D0AH	34	22H	LD (4016H), HL
32011	7D0BH	22	16H	
32012	7D0CH	64	40H	
32013	7D0DH	201	C9H	RET

Régi gép esetén az első két sort módosítani kell:

Cím		Tartalom		Mnemonik
32000	7D00H	33	21H	LD HL, 3078H
32001	7D01H	120	78H	

A gépi kódú programot vagy a monitorprogrammal, vagy – ha a programban használjuk – egy BASIC segédprogrammal tölthetjük be:

```
10 FOR I=32000 TO 32013: READ J : POKE I,J : NEXT
20 DATA 33,58,48,34,22,64,201,33,227,3,34,22,64,201
```

(A 20 sorban régi gépen 58 helyett 120-at kell írni!)

A 10 és 20 sorok beírása és futtatása után a billentyűismétlés és a tiltása is a szokott módon hívható. A hívás a BASIC programból is lehetséges, a futás nem áll meg. A billentyűismétlés hívása:
POKE 16526,0 : POKE 16527,125 : I=USR(0)

Az ismétlés tiltása:
POKE 16526,7 : POKE 16527,125 : I=USR(0)

Zátonyi Sándor

proper 16/W

PROP-GAZDA programcsomag

A GAZDA programcsomag a mezőgazdasági vállalatok ügyviteli és gazdálkodói tevékenységét támogató, számítógépre alapozott rendszer, amely átfogja az ügyviteli munka minden lényeges területét és naprakészen biztosítja a különböző szintű vezetők részére az ügyvitel területéről a szükséges információkat.
A rendszer számos számviteli, közgazdasági, mezőgazdasági és számítástechnikai szakember közös fejlesztési munkájának eredménye.
A GAZDA önálló alrendszerekből áll, amelyek lefedik

- főkönyvi könyvelés és utókalkuláció
- évközi eredmény-értékelés
- készletgazdálkodás
- bér- és munkaügy
- pénzügy
- állóeszköz-nyilvántartás

területén adatokat biztosítanak a gazdasági értékelésekhez.
Az egyes alrendszerek a felsorolt szolgáltatásaik összekapcsolásával együttesen egységes vállalati alkalmazási rendszert képeznek.
A rendszer nagy előnye, hogy rugalmasan adaptálható és bővíthető a vállalati gazdálkodás változó feltételeihez.

ÁLTALÁNOS ISMERTETÉS

A GAZDA rendszer a következő két programcsomagból áll:

- főkönyvelést integráló
- munkaszámos utókalkulációt integráló

programcsomagból.
Hangsúlyozzuk, hogy e programcsomagok szolgáltatásaiban jelentős számviteli és ökonómiai előrelépést jelentenek a szokásos ügyviteli rendszerekhez képest.
A rendszerre ráépülő alaprendszerek szolgáltatásainak kialakítása olyan, hogy azok képesek kiszolgálni a szükséges adatigényt.
Minden alaprendszernek szoros kapcsolata (adatforgalma) van a munkaszámos feldolgozással, de az alrendszerek egymás közti kapcsolata minimális.
Minden rendszerben elkülönül a pillanatnyi állapot nyilvántartása, amely egy tétel esetén másodperc elérés nagyságrendű. (Pl. 711–0012 főkönyvi számla pillanatnyi állapota)
A tételsoros kimutatások – amelyek „karton" szemléletben időrendi sorrendben tartalmazzák a gazdasági eseményeket – esetenként feldolgozással állíthatók elő, perc nagyságrendű eléréssel. (Pl. 711–0012 számla tételsorainak kigyűjtése)
A költség, az árbevétel és a naturális adatok gyűjtése üzemi belső munkaszámrendszerre történik. Ezáltal a rendszer független a számlakeret változásaitól, termelőegységek szintjén jobban kiszolgálható.
A kialakított rendszerkapcsolat és a munkaszámos gyűjtések adattartalma tág lehetőséget ad újabb és újabb rendszerek moduláris illesztésére a meglevő rendszerhez. (Ilyenek pl. a naturális vetítési adatok előfeldolgozásai segédüzemi, menetlevél-feldolgozó program keretében stb.)

A programok használata

A GAZDA programjai alapvetően párbeszédes, közvetlen adathozzáférést lehetővé tevő, képernyőorientált szemléletben készültek.
A rendszer az indítása után – a témakörnek megfelelően – a felhasználók mindennapos szakmai nyelvezetét követi, ÜZENET-eket küld és arra VÁLASZ-okat vár (minden esetben feltüntetve a lehetséges válaszok értékét).
A tömeges adatbevitel (bizonylatok feldolgozása) begépeléssel történik; a képernyő formailag követi a bizonylatok formátumát, így segítve az adatrögzítő munkáját.
A programok az adatbeviteleket követően a lehetséges szempontok alapján ellenőrzik az „elvi lehetségesség" kritériumát, hiba esetén módot adva a javításra.
A GAZDA alrendszerei azonos külső képernyőszerkesztési és kezelői beavatkozást megvalósító formában kialakítottak, így a gépkezelők átállási nehézség nélkül használhatják a különböző alrendszerek programjait.
Azonos tevékenységeket azonos megjegyzéssel ellátott üzenetben látnak és beavatkozniuk is ennek megfelelően kell.
Ha a feldolgozások időpontja egybeesik, a gép túlterheltsége csökkenthető azáltal, hogy a nagy tömegű adatrögzítés másik – esetleg kisebb kapacitású, olcsóbb – gépen is elvégezhető (off-line).
A rendszer lehetőséget ad a különböző gépeken rögzített és áthozott adatok feldolgozására (batch-kötegelt).

Fő jellemzők

A GAZDA összes szolgáltatásainak bevezetése esetén – átlagos ügyviteli méreteket figyelembe véve – a következő háttértárkapacitásokra van szükség:

	Mbyte
1. Anyagkönyvelés (10 000 cikkszám)	4
2. Bér- és munkaügy (1000 dolgozó)	3
3. Pénzügy (3000 ügyfél)	2
4. Állóeszköz nyilvántartás (15 000 állóeszköz)	1
5. Főkönyv és utókalkuláció (3000 számla)	1
(3000 munkaszám)	6
6. Gazdasági értékelés (100 szervezeti egység)	1
Összes adatterület	18
Programterület	2

A felsorolt összeállítás adattárolási igényei a tartósan lekötött lemezkapacitást mutatják. Erre a 27 Mbyte-os merevlemezek alkalmasak. A napi feldolgozásokhoz és a biztonsági mentésekhez a PROPER–16 gépek hajlékony mágneslemezes tárai használhatók. A megfelelő gyorsaságú nyomtató is nagyban növeli az átbocsátó képességet.

FŐKÖNYVI KÖNYVELÉS ÉS UTÓKALKULÁCIÓ

A GAZDA főkönyvi könyvelése és utókalkulációja a rendszer központi eleme. Ennek használata a technikai megoldás következtében kötelező.
A főkönyvi könyvelésben lehetőség van

- közvetlen számlaforgalmak könyvelésére egy és több könyvelési tétel összevonásával
- munkaszámos kigyűjtést egyidejűleg elvégző költséghely-költségviselői könyvelésre
- más rendszerek feladásainak kötegelt feldolgozására.

A számlaforgalomról naprakész, tetszőleges időpontokban kérhető főkönyvi kivonat.
Az utókalkuláció feladata kettős. Egyrészt a begyűjtött költségszerkezetből kiindulva a megfelelő utókalkulációs séma szerinti utókalkulálás.
Másrészt az elsődlegesen lerakott költségszerkezetben a költségátételezések és költségfelosztások elvégzése, főkönyvi vonzatának egyidejű kezelésével. Havonként az alaprendszerek felé előállítja a létező munkaszámokkal megnyitott, kigyűjtésekre szolgáló adatterületeket.
Az utókalkuláció szerkezete olyan, hogy valamennyi alaprendszerben munkaszámra kontírozás történjen.
A munkaszámok a mögötte levő költségszámlák, hozam, önköltség és árbevételszámlák megfelelő egymáshoz rendelése már a feldolgozás keretében történik. Ezáltal a széles körű termelői, felhasználói kör mentesül az üzemi konkrét szár ılatükör alkalmazásától.
A termelési folyamatok lezárásakor elvégzi a részletezett költség, árbevételi és naturális ráfordítási adatokból a termékek és a szolgáltatások utókalkulációját, eredményszámítását.

ÉVKÖZI EREDMÉNY-ÉRTÉKELÉS

A GAZDA az ügyviteli feldolgozások mellett nagy hangsúlyt fektet a gazdasági helyzet értékelésére.
Ez a szolgáltatás az utókalkuláció részére gyűjtött munkaszámos költségszerkezetből és az éves terv, bázis adatállományból indul ki.
Szervezeti egységenként az utókalkulációs munkaszámtípusok (költségszámítási sémák) bontásában összesített költségszerkezet, árbevétel, eredmény alakulást ad.
A kiértékelés tartalmazza a tervhez, bázishoz való viszonyítást és A/4 formátumban, áttekinthető információt ad a vezetők számára.
A szervezeti felépítésnek megfelelő összesítések révén a vezetés különböző szintjeihez differenciált információ hozható létre.

proper 16/W

Felvilágosítást ad:
Sci–L
Vevőszolgálat
1011 Budapest
Iskola utca 10.
Telefonszám: 260-000
Telexszám: 22-4590

A rendszer feltételezhetően újdonságnak számító szolgáltatása az önköltségszámítás szabályai szerinti – a költségszerkezet elemi költségnemekre történő – lebontása. Ennek keretében az általános költség, a segédüzemi és a fenntartóüzemi szolgáltatások elemi költségnemeikre bontva kerülnek a végterméknél kimutatásra.
Pl. a tehenészet költségszerkezetében a munkabér költségre a közvetlen munkabér mellett a segédüzemi és általános költség arányos munkabére is összesíthető.
Úgy gondoljuk, hogy a végtermék-kibocsátás valós számviteli adatokból kiinduló, ilyen értelmű költségszerkezeti kimunkálása nagyban segítheti a gazdasági vezetés munkáját.

KÉSZLETGAZDÁLKODÁS

A GAZDA készletgazdálkodása közös üzemi cikktörzsállomány nyilvántartásra alapozott, többraktáros, raktáronkénti forgalom feldolgozású, elszámolóáras könyvelési rendszer.
A közös cikktörzsállomány olyan funkcionális „kereső, válogató listákat" ad, amelyek minőségileg javíthatják a készletgazdálkodás szervezettségét.
A raktáránkénti forgalom-feldolgozás folyamatosan naprakész készletállapot nyilvántartást tesz lehetővé, elvégzi a főkönyvi feladások kigyűjtését, a készletfelhasználások munkaszámonkénti kimutatását.
Tételesen aktivizálja a forgalmi könyvelés adatait, ebből raktáránkénti, cikkenkénti kimutatásokat, idősoros és mozgásnemenkénti lekérdezéseket állít elő.
A begyűjtött főkönyvi forgalom és munkaszámos felhasználás adataiból a fogadórendszer részére közvetlenül felhasználható mágneses adathordozót készít.
Havi szinten biztosítja a bevételek likvidációit a számlaforgalom alapján, elszámolóár különbözet abszolút értékének számítását, felosztását a felhasználásra.
Funkcionális listák és kiértékelések támogatják a leltározási munkát.

Jellemző paraméterek

cikkszám: 10 jegyű numerikus ajánlott maximum 30...40 000
munkaszám: 1...2000
mozgásnem: 100-féle
költségnemi bontás: 30 db/5-ös számlára
raktáránkénti kapacitás: 6000 cikkszám (800 kbyte hajlékony lemez esetén)
használható bizonylat: 7 tételes, a rendszerhez kialakított.

BÉR- ÉS MUNKAÜGY

A GAZDA munkaügyi és bérelszámolási alrendszer a kialakított és folyamatosan karbantartott törzsadatállományok, valamint a havi bérszámfejtés végrehajtásához szükséges munkaidő- és teljesítményadatok felhasználásával készíti el a dolgozók munkabérének számfejtését és listázását.

1. A kialakított törzsadatállományok

- Dolgozói törzsadat
- Dolgozó járandósági törzsadat
- Dolgozó átutalás nélküli levonásos törzsadat
- Dolgozó átutalásos levonás törzsadat
- Szervezeti egységkód törzsadat
- Munkaszámos törzsadat
- Munkaköri törzsadat

2. A forgalmi adatok

- Bérszámfejtőlap, óra alapján történő bérfelosztásnál
- Bérszámfejtőlap, érték alapján történő bérfelosztásnál
- Esetenkénti járandóságok és kifizetések adatai
- Esetenkénti TB járandóságok és kifzetések adatai

3. A rendszer szolgáltatásai

- Dolgozók munkabérének számfejtése és kifizetési bizonylat elkészítése
- Dolgozó egyéni bérnyilvántartásának elkészítése
- Dolgozó egyéni tartozási és levonási nyilvántartásának elkészítése
- Munkabérek főkönyvi feladása munkaszámonként, tagalkalmazott bontásban.
- Kötelező havi, negyedéves, éves munkaügyi- és bérstatisztikák elkészítése.
- Gazdálkodási egység – rendszeres és szükség szerinti – munkaügyi és bérgazdálkodási kimutatások elkészítése
- Társadalombiztosítási járulék számítása, táppénz, ill. betegségi segély nélkül és szükséges TB statisztikák elkészítése

Az alrendszer a táblázatok elkészítésén kívül a törzsadatállományokba, valamint az egyéni nyilvántartásokba való – szükség szerinti – betekintést, a lekérdező rendszerben, képernyőn keresztül is lehetővé tesz.

PÉNZÜGY

A GAZDA pénzügyi alrendszere számlakészítési, aktív és passzív folyószámla könyvelési modulokból áll. A vevőket-szállítókat közös „ügyfél" adatállományként tartja nyilván.
A számlakészítés alapbizonylatokról a tételsoronkénti adattartalmak felvitelével készül.
Az aktív folyószámla-könyvelésben a kibocsátott számlák és a pénzügyi teljesítések alapján naprakészen vezeti a tételes folyószámlát, a nyitott tételek kivonatát. Ebből havonta főkönyvi feladás készül, munkaszámos-bontásban.
A passzív folyószámla-könyvelés a tételes szállítói folyószámlát vezeti, a nyitott tételeket tartalmazó kivonatot előállítja. A szállítói számlákat egy átvezetési számlával szemben könyveli és közben könyvelési segédletet készít a kontírozáshoz.
A rendszer a napi pénzügyi helyzetet tükröző rövidtávú pénzügyi helyzetértékelő „napi jelentésekkel" segíti a gazdasági vezetés munkáját.
A rendszer a későbbiek során adózási és pénzügyi statisztikai modulokkal bővül.

ÁLLÓESZKÖZ-NYILVÁNTARTÁS

A GAZDA rendszerében az állóeszköz alrendszer egyedi nyilvántartás, értékcsökkenés elszámolás, munkaszámos feladás, leltárkiértékelés, funkcionális keresések és statisztikai adatszolgáltatás modulokból áll.
Egyedi nyilvántartás keretében az állóeszköz üzembehelyezéséből a teljes használati időt végigkövetve nyilvántartja a gazdasági eredményeket.
Értékcsökkenés elszámolása keretében havi (vagy negyedévi) időszakonként tömeges feldolgozás keretében elvégzi az értékcsökkenés számítást, a nettó értékek módosítását, ezzel összefüggő főkönyvi feladási tételeket mágneses adathordozón előállítja.
Munkaszámos feladás célja, hogy a számfejtett értékcsökkenés munkaszámonkénti terhelésre kerüljön, az utókalkuláció számára feldolgozható formában.
Leltárkiértékelés keretében a felvételezést megelőző lekérdezések és a feldolgozás kiértékelései szerepelnek, kimunkálva az eltéréseket a főkönyvi kontírozási segédlet egyidejű elkészítésével.
A funkcionális keresések segítik az állóeszközgazdálkodás pillanatnyi helyzetének áttekintését. Egyedi szám, leírási kulcs, statisztikai besorolás, szervezeti egység stb. szerint egyedi vagy tömeges lekérdezéseket biztosít képernyőre, nyomtatásra.
Statisztikai kiértékelések kérhetők a külső adatszolgáltatási kötelezettségek gyors, szerkezetében az igényeket követő formában.

SZOFTVER-, ILL. HARDVERKÖRNYEZET

- 64 kbyte RAM
- 25x80 karakteres kijelző, billentyűzet
- 27 Mbyte merev lemezes tár
- hajlékony mágneslemez meghajtó egység
- nyomtató

A programcsomag a PROPOS–16 vagy ezzel kompatibilis operációs rendszer alatt működik.

POSTA

BP. 1986

Tisztelt BIT-LET szerkesztőség!

Több kérdésem is lenne Önökhöz:

1. Jóformán minden számukban (és a magazinban is) találkoztam dr. Simonyi Endre velős cikkeivel. Most már szerintem, igazán szánhatnának egy „Vallatót" a SIMON 68-as gépre, hiszen e cikkekből azt lehet kivenni, hogy a COMMODORE és a SINCLAIR csak Micimackó hozzá képest.

2. Elolvastam Szőke Ferenc szenzációs PRIMO pótvallatását. Én, az önök helyében fölvenném az Inkvizítorok közé. De elolvastam a VC–20 vallatását is. Előzetesen be kell vallanom, hogy fanatikus ZX-párti vagyok (főleg SPECTRUM) és VC–20 előtt kb. 2 órát ültem. Azonban egyszerűen nem tudom megérteni, hogy a következő kínoknál miért szerepelt rosszabbul a VC–20 a C–64-nél. (Előítélet?!)

– perifériák (2. kín)
– kazettás tárolás (5. kín) – holott

a géphez (mint írják) ugyanúgy csatlakoztatható floppy, és magnó (mindkettő azonos a C–64-esével), mint a C–64-eshez.

– billentyűzet (8. kín)
– editálás (10. kín) idézem Önöket:

„A gép editora azonos a C–64-esével!"

*Ugyanebben a lapban a 11. TV BASIC hasábjain közölt táblázathoz azt írják, hogy a SIMON'S BASIC belső függvényeit *-gal jelölik. Sajnos a táblázatban egy fia csillag nem sok, de annyit se találtam, pedig tűvé tettem érte. (Egyébként a SPECTRUM-ra készült Beta BASIC – kiegészítő belső függvényeit is beírhatták volna. (Igaz, akkor jóval nagyobb táblázat kellett volna!)*

Iskolánkban egy pár SPECTRUM-os segítségével és részvételével – „őszinte részvétel", mondták először ötletemre – ZX-klubot alapítottunk. Ezt szeretnénk kiterjeszteni a „külvilág" elé is. Ezért most valamilyen helyet keresünk a belvárosban, mert a suliban ez nem valósítható meg. Ha valaki tudna segíteni...!

Most készült el gépi kódú kódoló-dekódoló programunk, mellyel több képernyőtartalmat rakhatunk el, majd ezután gyorsan (vagy lassan) visszahívhatók. Így pl. forgatások (reklámhoz) valósíthatók meg. Ha érdekli Önöket, kérem, valamilyen módon válaszoljanak! (Mintát is tudunk mutatni, már kész a TV-BASIC és egy árnyékolt (Skála) forgatás, természetesen 3 D-ben.

Vállaljuk bármilyen célú program megírását BASIC-ben, és egy kicsit Assemblyben is. Telefonom: 311-431.

Jutasi Tamás, a Vági István ÉSZI III. osztályos tanulója

Tisztelt Olvasónk!

Röviden szeretnék válaszolni kérdéseire:

1. A SIMON'68 nem eléggé közismert ahhoz, hogy a Vallató alapkövetelményét teljesítse. Elvünk ugyanis, hogy csak nagy példányszámban Magyarországon létező gépről készítünk vallatót.

2. Az inkvizítorok „toborzása" elég széles körben történik. Nyilván sok tíz vagy száz embert érdemes lenne még meghívni, de mi kb. 10-et szoktunk. Utólagos hozzászólásokat szívesen veszünk, illetve közlünk.

A legelső Vallatónál közöltük, hogy ezt nagyon is szubjektív értékeléssé akarjuk tenni. A különbségek ebből adódnak. (Így lehet az is, hogy ugyanazon vallatónál egy kínra van, aki 5-öst, van aki 2-est ad. Pedig ugyanarról a dologról volt szó!)

Előítéletről tulajdonképpen csak pozitív értelemben beszélhetünk, hiszen ezek az emberek ezt a gépet használják, és ha nem szeretnék, akkor nyilván mást igyekeznének választani.

3. A TV BASIC rovat teljesen független a BIT-LET-től, az észrevételeit továbbítjuk hozzájuk.

4–6. Valószínűleg közzé fogjuk tenni a lapban, további választ nem igényel.

További sikereket kívánunk!

Olvastam, sőt le is pötyögtettem a múltkori számban (144.) található Finom grafika VC 20-on című programot. Jó lenne, ha közölnék a program gépi kódú változatát. Kérdésem: a VC 20 melyik biten tárolja a joystick (botkormány) helyzetét?

Csóka István, 9023 Győr, Felszabadulás út 95. fsz. 2.

Első kérésével kapcsolatban még némi türelmet kérünk. A második témára íme válaszunk: A botkormány állapotát a 37151-es és 37152-es regiszter egyes bitjei mutatják:

37151	2. bit =0	fel
37151	3. bit =0	le
37151	4. bit =0	balra
37151	5. bit =0	tűz
37152	7. bit =0	jobbra

A két regiszter olvasása (PEEK) előtt a 37154-es regiszter 7. bitjét be kell állítani. Ha ez a bit zérus, akkor olvashatjuk a 37152-es regisztert, ha pedig 1, akkor olvashatjuk a 37151-es regisztert.

A botkormányt lehetőleg csak program módban kezeljük, és a programból való kilépés előtt állítsuk vissza a regiszterek eredeti tartalmát:

37151	126
37152	247
37154	255

Igen tisztelt Szerkesztőség!

A BIT-LET márciusi számában olvastam a COMMODORE VC 20-ról készített cikküket. Ebben felvetették, hogy hiányzik a magyar nyelvű kézikönyv.

A kézikönyv magyar nyelvű fordítását saját kedvtelésemre elkezdtem, és jelenleg kb. 70 %-os készültségben vagyok. Megjegyzem, hogy a kézikönyvet szakmailag is átvizsgáltam és a gépet kipróbálva ellenőriztem annak kitételeit. Rendelkezésemre állnak továbbá a VC 20-ról olyan kézikönyvek, amelyek annak operációs rendszeréi és programozási alkalmazási módját a kézikönyvnél részletesebben tárgyalják. Módomban állt a gépen sok „ügyes" programot is kipróbálni.

Akinek a fentiek felkeltették az érdeklődését, kérem válaszoljanak, szívesen állok a felhasználók rendelkezésére.

Tóth László, Sopron, Baross u. 8/d. 9400

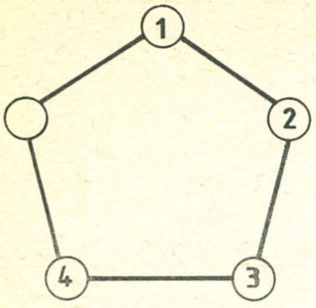
A számítógépes oktatás beindultával ma már hazánk minden közép- és felsőfokú oktatási intézményében megtalálhatók a személyi számítógépek, s köztük nem elhanyagolható a HT–1080Z típusúak a száma. Ezekkel kapcsolatos levelem.

Sokan vannak, akik jelenleg ismerkednek a számítástechnikával, „kóstolgatják azt"! Munkájukat szeretném megkönnyíteni egy programmal, amely a hibaüzenetet magyar nyelven adja. Bárkivel előfordulhat, hogy nem jut eszébe az, hogy például mit is takar a DD Error. Hát még a kezdők! Ez a program ezt a helyzetet szünteti meg.

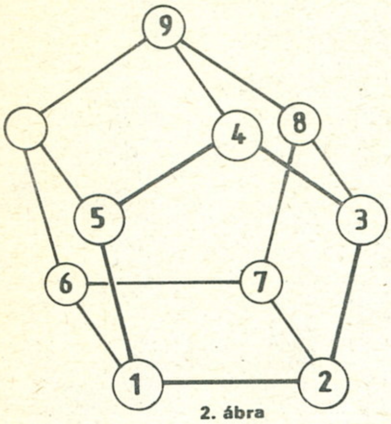
Éppen ezért, akik érdeklődnek a program iránt vagy szükségét érzik, küldjenek nekem egy kazettát válaszborítékkal, felcímezve, s ellenszolgáltatás kívánalma nélkül megkapják a szellemi terméket.

Pelhrimovszky Zsolt, Budapest, Gárdos Mariska u. 6. IV/24. 1032

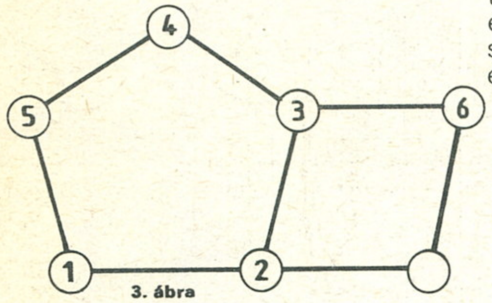
MEGJEGYZÉSEK A PRIMO-NYERŐHÖZ



1. ábra



2. ábra



3. ábra

Bár az értékelés még nem fejeződött be, mégis szeretnénk néhány értékelő szót szólni.

Az első feladatra még 223 megoldás érkezett, végül csak 71 olyan pályázónk akadt, aki mind a 3 feladatra küldött be megoldást.

Legkönnyebbnek az első feladat bizonyult, a beérkezett megoldások majdnem mind helyesek; elég sokan küldtek be programot is, melyre 2 plusz pontot kaptak (a sorsoláson mindenki részt vesz, aki eléri a 8+10+12=30 pontot, a plusz 2 pont „csak" arra jó, hogy egy másik feladatnál elveszített pontot pótolja). A második feladatra már lényegesen kevesebb megoldás érkezett, de ezek nagy része is helyes. A legnehezebb a 3. feladat volt, erre nagyon sok hibás bizonyítás érkezett, sőt 1–2 rossz válasz is. Sokan észrevették, hogy a játék hasonlít a Sam Loyd-féle 15-ös játékra, ahol szintén nem lehet két kis négyzetet megcserélni. Akik ismerték ennek bizonyítását, azok egy kis okoskodással kiterjeszthették a feladatban szereplő esetre is. Ez azonban nem mindenkinek sikerült helyesen.

A típushiba a következő volt:
„Mivel a feladat állítása a 15-ös játékra igaz, ezért igaz a térbeli esetben is, hiszen két kockán kívül minden a helyén marad, ezért ha egy másik (nem ezeket tartalmazó síkba) egy kockát beviszünk, azt ugyanott vissza is kell vinni."
Ennek a gondolatmenetnek a tévességét jól illusztrálja a következő példa:
Legyen először a játékmezőnk egy szabályos ötszög (I. 1. ábra), melynek négy csúcsában van 1–1 számozott korong. Itt nyilván nem lehet elérni, hogy csak az 1-es és a 2-es korong legyen felcserélve (a korongokat itt az ötszög élein tologathatjuk).

Legyen egy másik játékunk a 2. ábra szerinti, azaz egy ötszög alapú hasáb élei mentén tologathatunk. Az idézett gondolatmenet alapján itt sem lehetne elérni, hogy csak 2 korong legyen felcserélve; ez azonban nem igaz, most van olyan tologatássorozat, melynek eredményeképpen minden a helyén marad, csak az 1-es és 2-es korong lesz felcserélve. Ennek megkeresése szorgalmi házi feladat, segítségképpen eláruljuk, hogy mindezt már a 3. ábra szerinti játékban is el lehet érni.



A szakköri pályázat 2. feladata egy könnyen programozható játék elkészítése a HT–1080 Z gépen.
A játék a következő: van 10 db négyzetünk, melyeknek egyik oldala fehér, a másik fekete. Ezek le vannak rakva sorban az asztalon, némelyik a fehér, mások a fekete oldalukkal felfelé. Két játékos felváltva fordíthat meg négyzeteket a következő szabály szerint:
a soron következő megfordíthat egy olyan négyzetet, melynek a fehér oldala van felfelé, vagy
megfordíthat 2 különböző négyzetet is, de akkor a jobb oldalit a fehér oldaláról a feketére kell fordítania.
A vesztes az, aki már nem tud fordítani, tehát az nyer, akinek fordítása után az összes négyzet a fekete oldalával felfelé van.

Kérjük levágni és a levélre felragasztani! Beküldési határidő június 15.

Mielőtt bárki nekiesne a program megvalósításának, alaposan el kell gondolkodnia a játékon, s meg kell határoznia az általános nyerő stratégiáját. Ugyanis a feladat az, hogy írjunk egy programot, mely véletlenszerű kezdőhelyzetet teremtve olyan jól játszik a kezelőjével szemben, hogy hacsak lehetséges, akkor nyer.

BIT-LET Szerkesztőségünk címe: ÖTLET Budapest 1986. Szerkesztő: Angyalosi László. Művészeti szerkesztő: Pribelszky Pál. Tördelő: Domokos Imre. Hírrovat: Csányi György. Vallató: Pogány György. Sorvezető: Székely Jenő. Posta: Halász Péter